Netflix Data Analysis Overview

- This project looks at the Netflix dataset to find useful insights about the types of content, popular genres, countries producing content, rating trends, how content has changed over time, and what users like.
- We use Python tools like Pandas, Seaborn, and Matplotlib to clean the data and create visual charts.
- The dataset shows Netflix's content from 1925 to 2021. Our main goal is to find patterns that can help with planning content and keeping users more engaged.

- By: Neetant

Dataset Preparation & Cleaning

➤ **Loaded Netflix Data**
We added the **netflix_titles.csv** file to start our analysis.

➤ **Cleaned the Data**
We removed extra columns like **show_id**, **director**, **cast**, and **description** to keep only the important ones.

➤ **Filled Missing Data**
We replaced missing values with **'Unknown'** to complete the dataset.

➤ **Fixed Date Format**
We changed the **date_added** column to the same date format for all rows.

➤ **Separated Genres**
We split the **listed_in** column into individual genres and gave each genre its own row.

```
df = pd.read_csv(r"C:\Users\dell\Desktop\neetant\netflix_titles.csv")
```

```
df
```

```
df.isnull().sum()
```

```
show_id         0
type            0
title           0
director     2634
cast          825
country       831
date_added     10
release_year    0
rating          4
duration        3
listed_in       0
description     0
dtype: int64
```

```
print("\nMissing values before cleaning:\n", df.isna().sum())

df.drop_duplicates(inplace=True)

df['country'] = df['country'].fillna(df['country'].mode()[0])
df['rating'] = df['rating'].fillna(df['rating'].mode()[0])

df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
df.dropna(subset=['date_added'], inplace=True)

df['director'].fillna('Unknown', inplace=True)
df['cast'].fillna('Unknown', inplace=True)


print("\nMissing values after cleaning:\n", df.isna().sum())
print("Cleaned shape:", df.shape)
```

Dataset Overview

- Total entries: 8807
- Rows- 6000
- Columns- 12(show id , type , title , director , cast , country , date added , release year , rating , duration , listed in , description)

```
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

Library Used & Descriptive Statistics:

- **Pandas** – Data manipulation
- **NumPy** – Numerical operations and handling missing data
- **Matplotlib pyplot** – Basic charting
- **Seaborn** – Statistical and elegant visualizations
- **Plotly Express** – Interactive plots

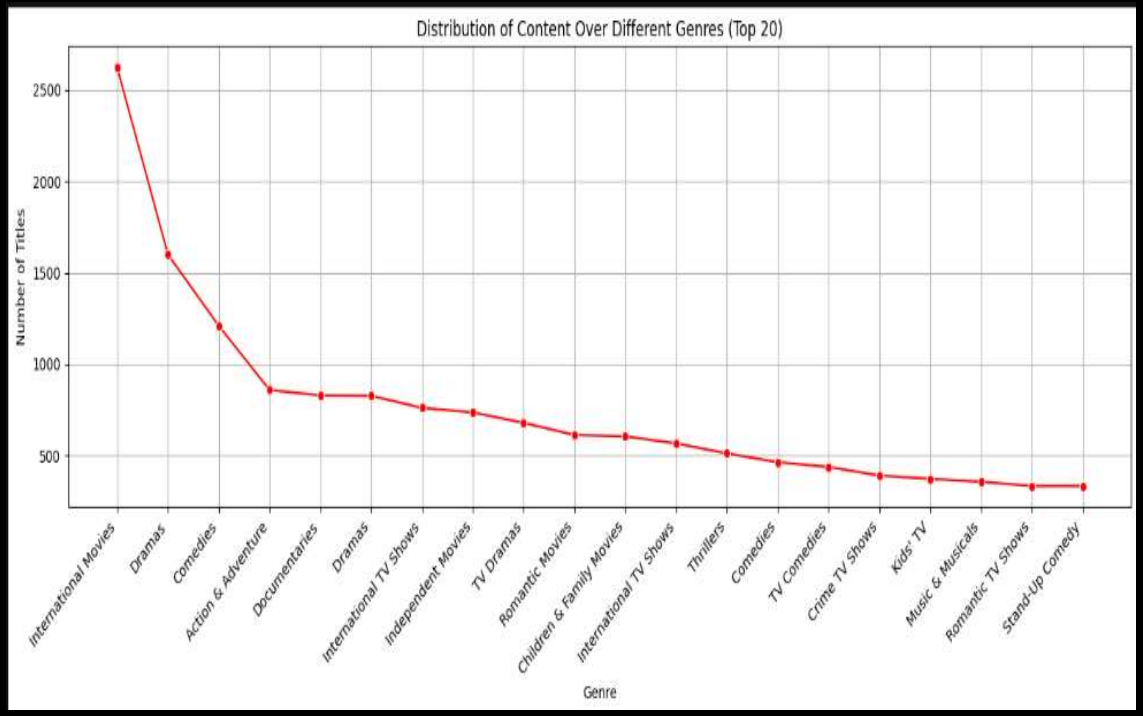
```
import pandas as pd
import numpy as np
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df.describe()
```

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

Data Visualization - Part 1: Create visualizations to represent the distribution of content over different genres

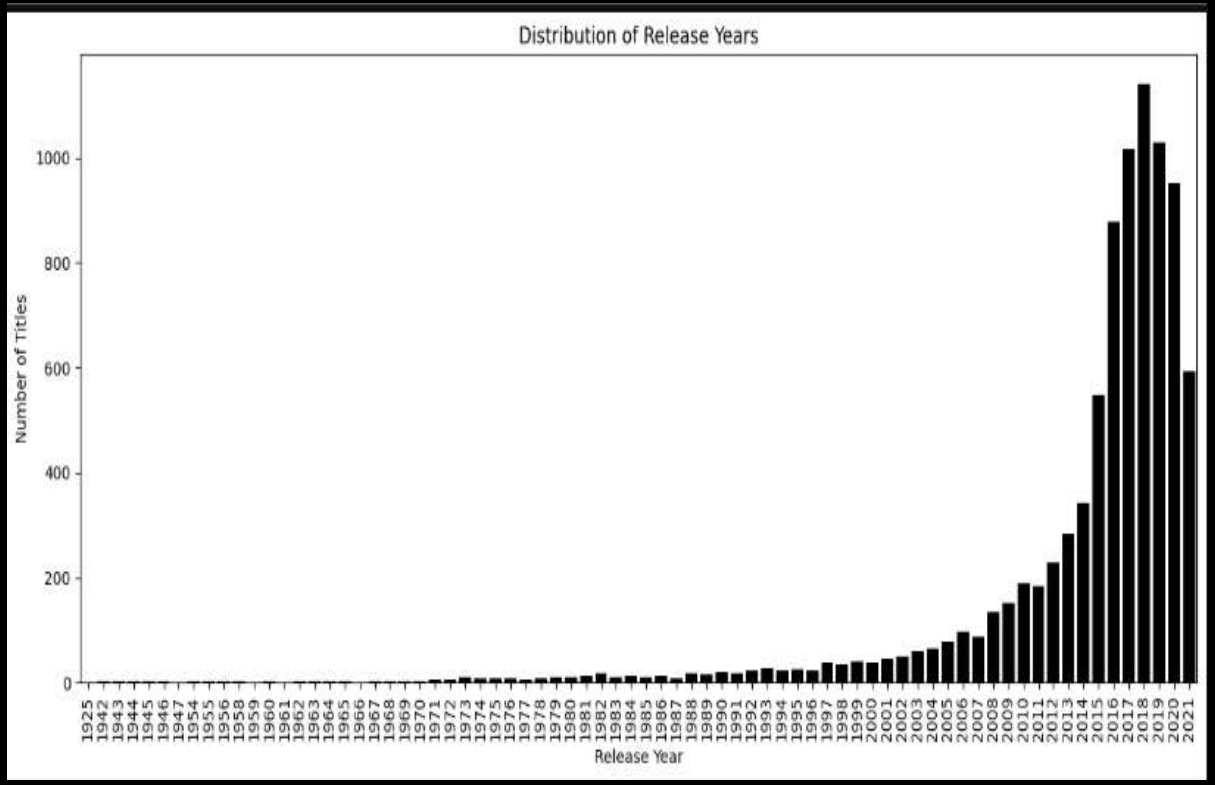
```
gen= df['listed_in'].dropna().str.split(',')
exp= gen.explode()

counts= exp.value_counts()
plt.figure(figsize=(14, 6))
sns.lineplot(x=counts.index[:20], y=counts.values[:20], marker='o', color='red')
plt.xticks(rotation=45, ha='right')
plt.title('Distribution of Content Over Different Genres (Top 20)')
plt.xlabel('Genre')
plt.ylabel('Number of Titles')
plt.grid(True)
plt.tight_layout()
plt.show()
```



- Replace any missing values (NaN) with **"Unknown"** to ensure all entries have at least one genre.
- **Split Genre Strings into Lists**
- Convert genre strings like "Drama, Comedy" into lists: ["Drama", "Comedy"].
- **Flatten Genre Lists with .explode()**
- Turn each list into separate rows so that each title is counted once per genre—this avoids miscounting multi-genre titles.
- **Clean the Genre Field (listed_in)**
- **Count Genres & Select Top 20**
- Use value counts () to find how often each genre appears. Then pick the **top 20 most frequent genres** for visualization.
- **Plot Top 20 Genres**
- Use **Seaborn's barplot()** with a "hot" color palette to show the genre counts in a clear, visually appealing bar chart.
- **Visualize Content Types (Movies vs TV Shows)**
- Use Seaborn's countplot () on the type column to show how many titles are **Movies** vs. **TV Shows**.

Data Visualization - Part 2: Visualize the distribution of content across release years

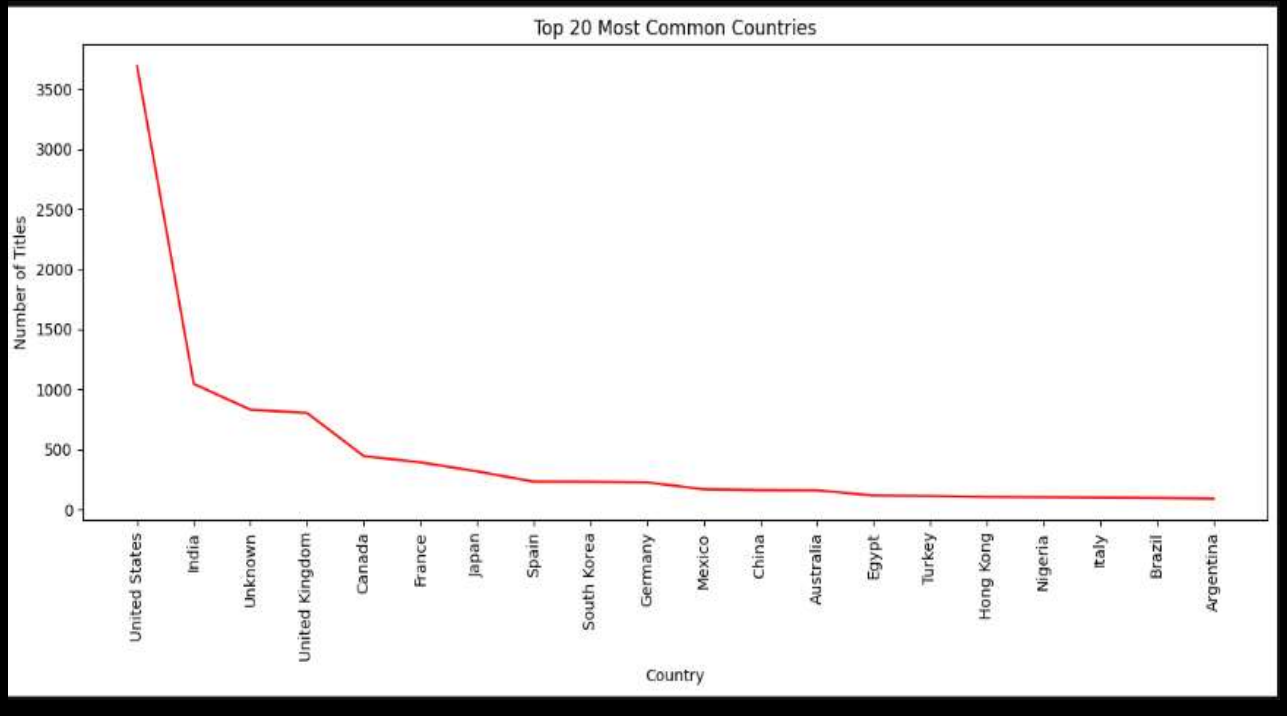
```
release_year_counts = df['release_year'].value_counts().sort_index()
plt.figure(figsize=(12, 6))
sns.barplot(x=release_year_counts.index, y=release_year_counts.values, color='black')
plt.title('Distribution of Release Years')
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



The analysis focuses on the **release year** column, using .value counts() to count how many titles were released each year and then sorting these counts in ascending order with .sort index(). A **Seaborn bar plot** is created to display the number of titles per year, and the x-axis labels are rotated by 90° to prevent overlap. The use of plt.tight layout() ensures that all chart elements—like labels and axes—fit neatly without crowding. This visualization makes it easy to see how Netflix's content output changes over time, highlighting trends such as growth in recent years

Data Visualization - Part 3: Explore the geographical distribution of content (if applicable)

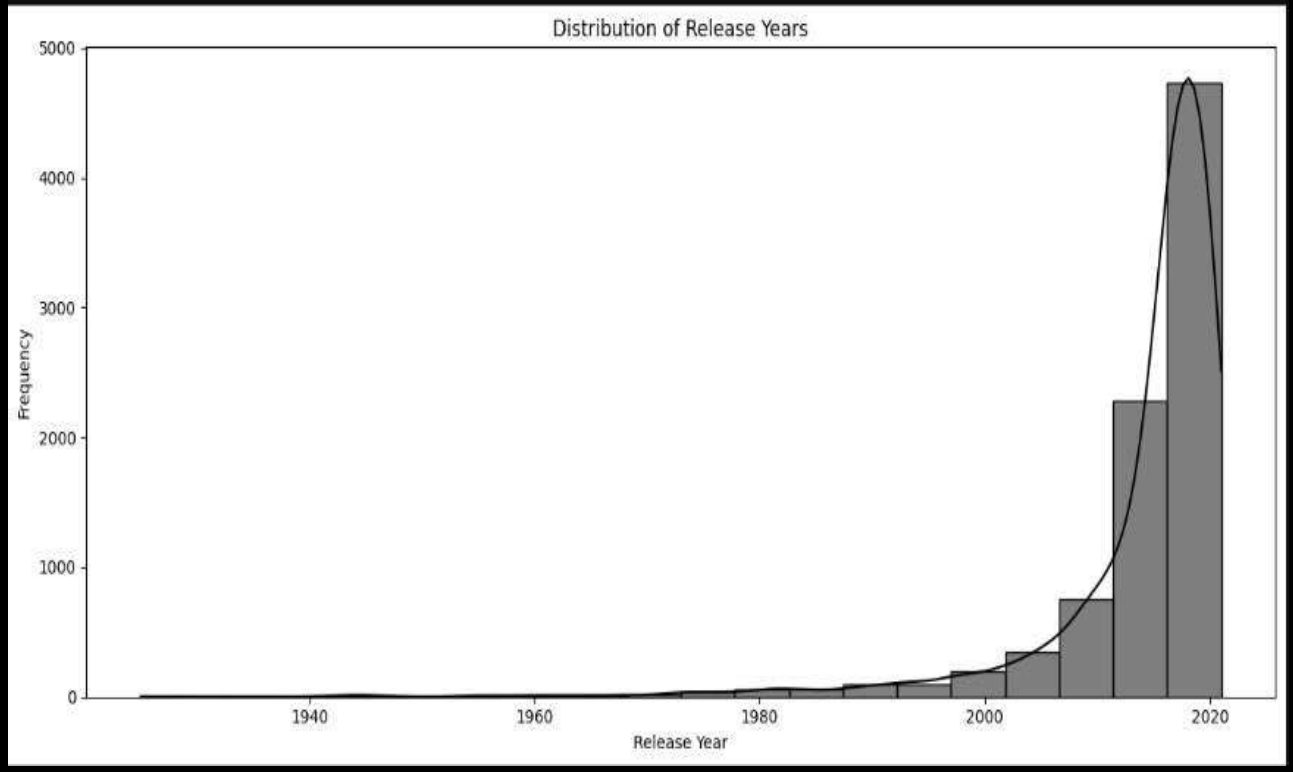
```
contet_series = df['country'].dropna().str.split(', ')
country_exploded = contet_series.explode()
country_counts = country_exploded.value_counts()

plt.figure(figsize=(12, 6))
plt.plot(country_counts.index[:20], country_counts.values[:20],color = "red")
plt.title('Top 20 Most Common Countries')
plt.xlabel('Country')
plt.ylabel('Number of Titles')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



- **Objective**: Analyze country-wise distribution of Netflix content.
- Remove missing values using .dropna().
- Split entries with multiple countries using .str.split(', ').
- Use .explode() to create a separate row for each country.
- Select the top 20 countries with the most titles.
- Use matplotlib to create a line plot of these top 20 countries.
- **Result**: Visualizes which countries produce the most Netflix content.

Time Series Analysis: If there's a temporal component, perform time series analysis to identify trends and patterns over time.

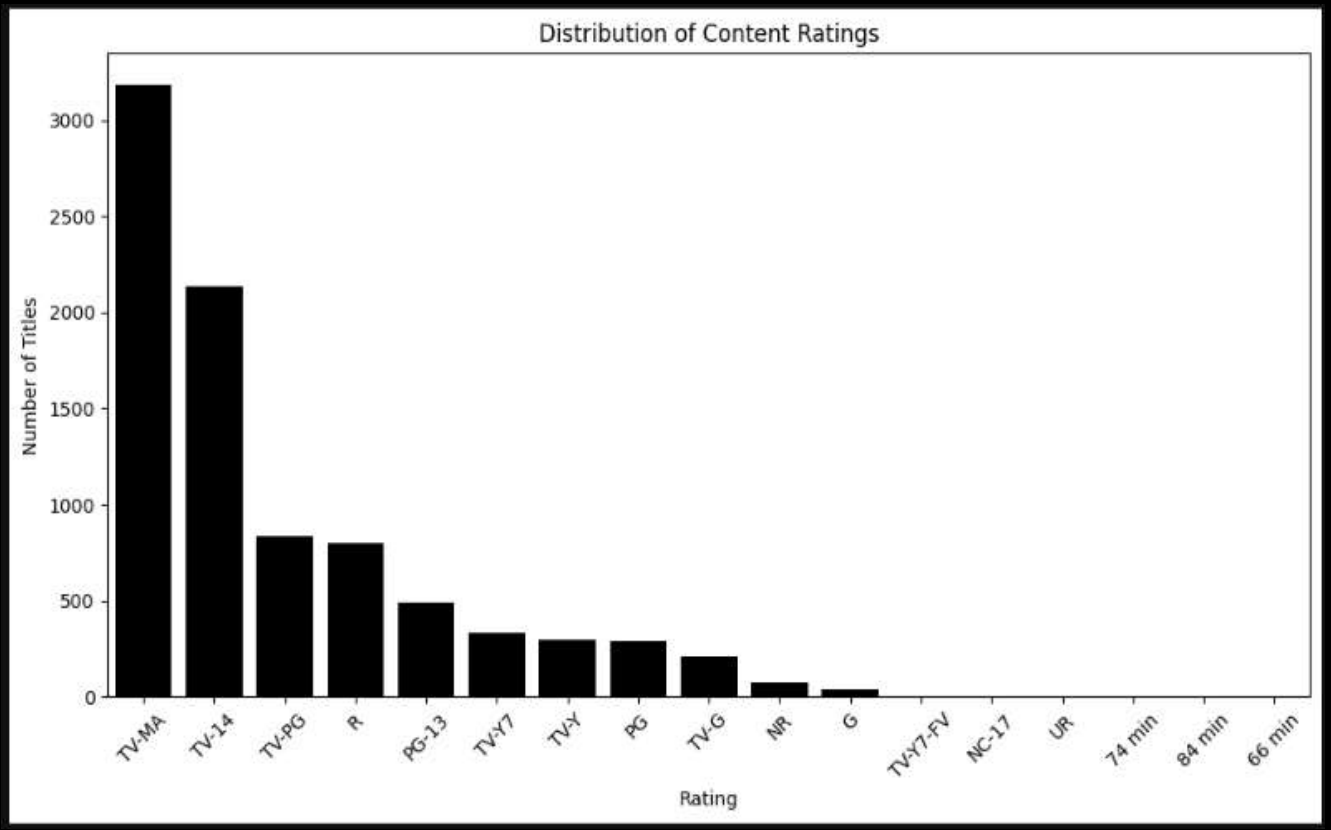
```
plt.figure(figsize=(12, 6))
sns.histplot(data=df, x='release_year', bins=20, kde=True, color='black')
plt.title('Distribution of Release Years')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



- **Purpose**: Creates a histogram to show the distribution of Netflix titles by release year.
- **Bins**: Uses **20 bins** to group release years into intervals.
- **KDE Curve**: kde=True adds a smooth line to show the overall trend.
- **Layout**: tight_layout() used for spacing and clarity.
- **Insight**: Helps understand how content release trends have changed over the years

Content Analysis - Part 1: Analyze the distribution of content ratings.

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='rating', order=df['rating'].value_counts().index,color = "black")
plt.title('Distribution of Content Ratings')
plt.xlabel('Rating')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



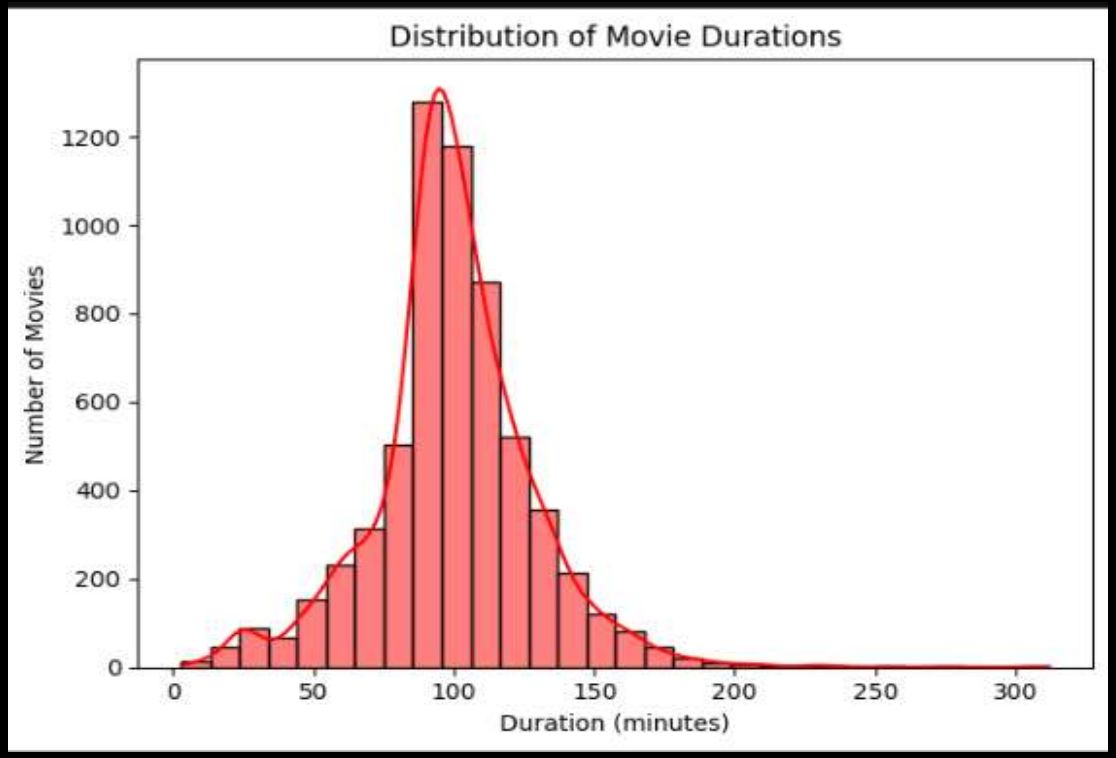
- ✓ **Purpose**: Visualize the distribution of content ratings in the Netflix dataset.
- ✓ **Data Source**: a['rating'] column from the DataFrame.
- ✓ **Order**: Bars are sorted in descending order using a['rating'].value_counts().index.
- ✓ **Rotation**: X-axis labels rotated 45° for readability.
- ✓ **Layout**: plt.tight_layout() used to prevent overlapping.
- ✓ **Insight**: Shows which content ratings are most common on Netflix.
- ✓ **Usefulness**: Helps understand Netflix's audience and content classification.

Content Analysis - Part 2: Explore the length of movies or episodes and identify any trends.

```
movies = df[df['type']=='Movie'].copy()
tv_shows = df[df['type']=='TV Show'].copy()
```

```
movies['duration_minute'] = movies['duration'].str.extract('(\d+)').astype(float)
tv_shows['duration_minute'] = tv_shows['duration'].str.extract('(\d+)').astype(float)
```

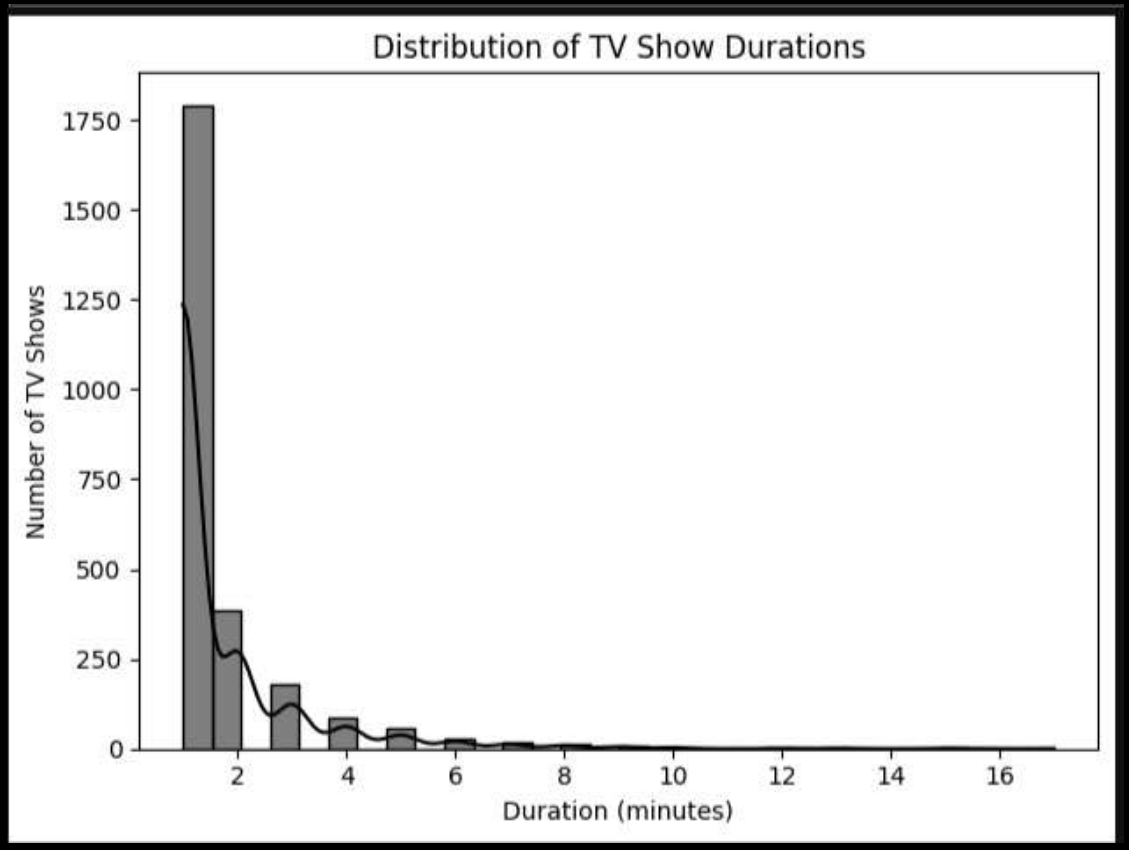
```
sns.histplot(movies['duration_minute'], bins=30, kde=True, color='Red')
plt.title('Distribution of Movie Durations')
plt.xlabel('Duration (minutes)')
plt.ylabel('Number of Movies')
plt.tight_layout()
plt.show()
```



- The code analyzes movie durations from the Netflix dataset.
- From the 'duration' column (e.g., "90 min", "2 Seasons"), only the number is extracted using str.extract('(\d+)').
- The extracted number is converted to **float** and stored in a new column called 'duration_minute'.
- A histogram with **KDE (Kernel Density Estimate)** is plotted for movie durations using seaborn.histplot.
- The plot helps identify common movie durations on Netflix (e.g., most movies being around **90 minute**

Content Analysis - Part 2: Explore the length of movies or episodes and identify any trends.

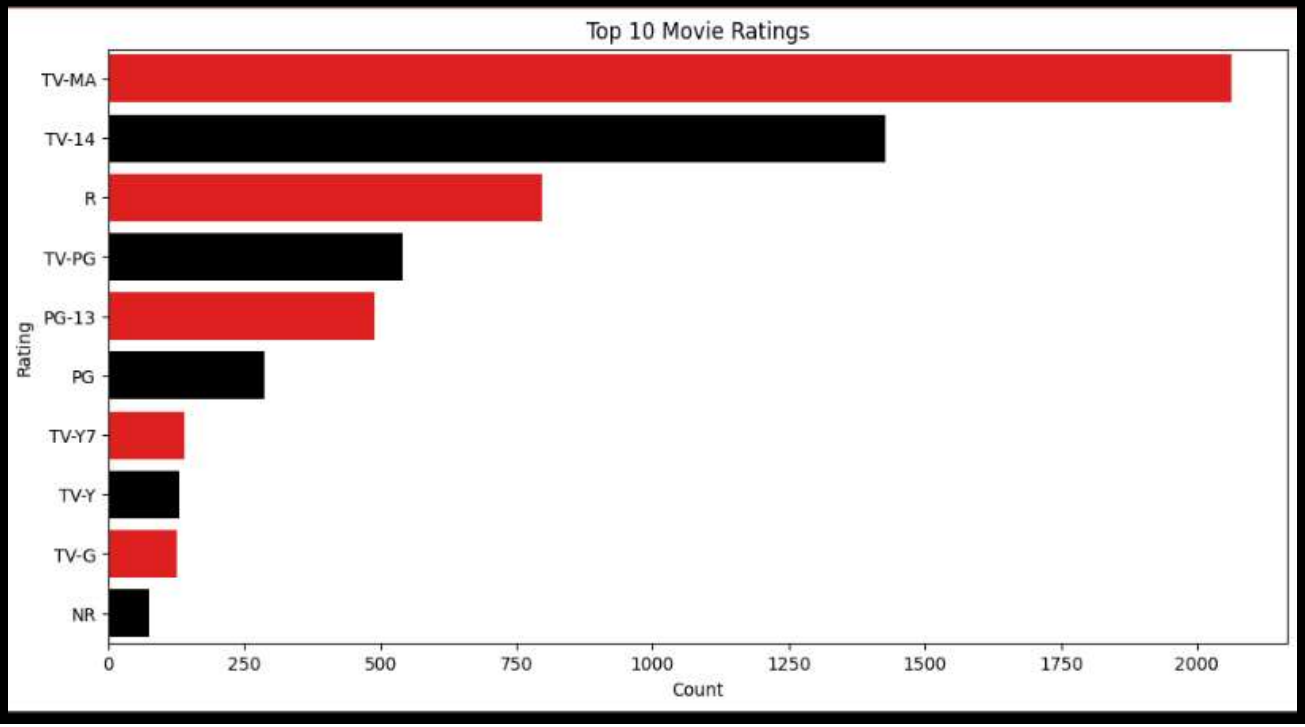
```
sns.histplot(tv_shows['duration_minute'], bins=30, kde=True, color='black')
plt.title('Distribution of TV Show Durations')
plt.xlabel('Duration (minutes)')
plt.ylabel('Number of TV Shows')
plt.tight_layout()
plt.show()
```



- The code visualizes TV show durations using a histogram.
- It uses Seaborn's histplot on the duration_minute column from the tv_shows dataset.
- The plot helps analyze whether most TV shows are short, medium, or long in duration on Netflix.

Top Lists and Recommendations: Identify and present top-rated movies or TV shows based on user ratings.

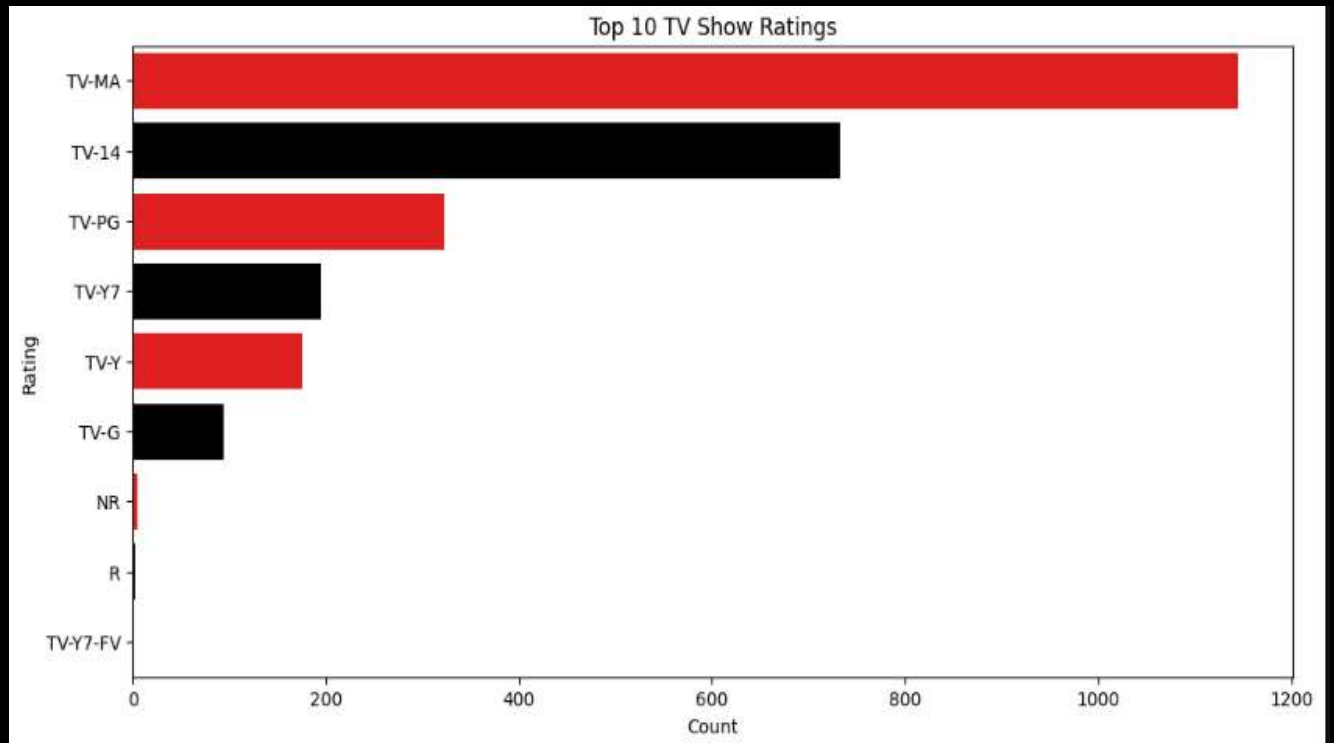
```
plt.figure(figsize=(12, 6))
sns.countplot(y='rating', data=df[df['type'] == 'Movie'], order=df[df['type'] == 'Movie']['rating'].value_counts().index[:10], palette=['re
plt.title('Top 10 Movie Ratings')
plt.xlabel('Count')
plt.ylabel('Rating')
plt.show()

plt.figure(figsize=(12, 6))
sns.countplot(y='rating', data=df[df['type'] == 'TV Show'], order=df[df['type'] == 'TV Show']['rating'].value_counts().index[:10], palette=
plt.title('Top 10 TV Show Ratings')
plt.xlabel('Count')
plt.ylabel('Rating')
plt.show()
```



- **Most Common Movie Ratings:**
- You can clearly see which ratings (e.g., **TV-MA**, **PG-13**, **R**) are most frequently assigned to Netflix movies.
- Typically, **TV-MA** and **PG-13** dominate, reflecting mature and teenage-friendly content.
- **Audience Targeting:**
- A high count of **mature ratings** (TV-MA, R) suggests a focus on adult audiences.
- Fewer **G** or **TV-Y** ratings might indicate limited content for very young viewers.
- **Platform Strategy Insight:**
- Knowing which ratings are most common helps understand Netflix's **content curation and age-based targeting**.
- Useful for parents, creators, and policy watchers alike.
- **Content Filtering Relevance:**

Top Lists and Recommendations: Identify and present top-rated movies or TV shows based on user ratings.

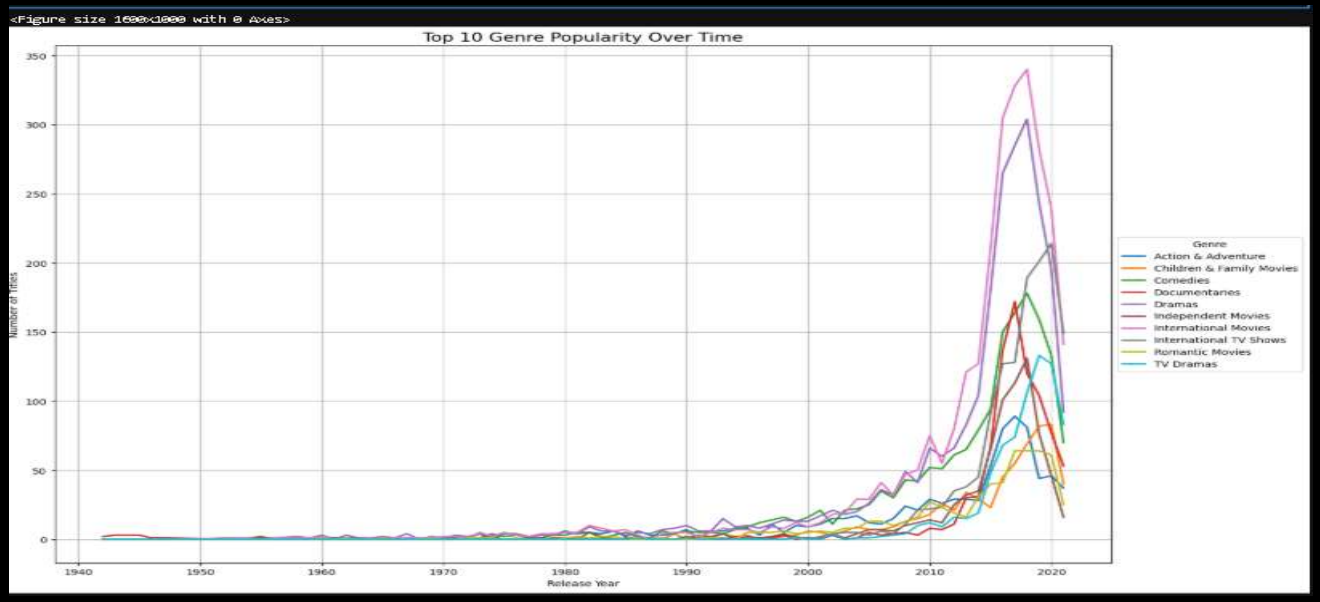
```
plt.figure(figsize=(12, 6))
sns.countplot(y='rating', data=df[df['type'] == 'Movie'], order=df[df['type'] == 'Movie']['rating'].value_counts().index[:10], palette=['re
plt.title('Top 10 Movie Ratings')
plt.xlabel('Count')
plt.ylabel('Rating')
plt.show()

plt.figure(figsize=(12, 6))
sns.countplot(y='rating', data=df[df['type'] == 'TV Show'], order=df[df['type'] == 'TV Show']['rating'].value_counts().index[:10], palette=
plt.title('Top 10 TV Show Ratings')
plt.xlabel('Count')
plt.ylabel('Rating')
plt.show()
```



- **Dominance of TV Ratings:**
- The most common ratings will likely be **TV-MA**, **TV-14**, and **TV-PG**, which are standard for TV show content regulation.
- These ratings suggest a wide range of shows for both mature and teen audiences.
- **Family vs. Adult-Oriented Content:**
- A high count of **TV-MA** shows a focus on adult-oriented series (violence, mature themes).
- Lower counts of **TV-Y** or **TV-G** indicate fewer shows aimed at very young children.
- **Content Strategy Focus:**
- If mature ratings dominate, it suggests Netflix is investing more in **adult dramas, thrillers, or reality series**.
- If there's a balanced mix, the platform is likely targeting **diverse audience segments**.
- **Viewer Usefulness:**
- Helps parents, educators, and viewers quickly understand what kind of content dominates the platform's TV show library

Genre Trends: Analyze trends in the popularity of different genres over time.

```
genres = df[['release_year', 'listed_in']].dropna().copy()
genres['listed_in'] = genres['listed_in'].str.split(', ')
genres = genres.explode('listed_in')
top_genres = genres['listed_in'].value_counts().nlargest(10).index
genres_filtered = genres[genres['listed_in'].isin(top_genres)]
genres_trends = genres_filtered.groupby(['release_year', 'listed_in']).size().reset_index(name='count')
genre_pivot = genres_trends.pivot(index='release_year', columns='listed_in', values='count').fillna(0)
plt.figure(figsize=(16, 10))
genre_pivot.plot(kind='line', figsize=(16, 10), linewidth=2)
plt.title('Top 10 Genre Popularity Over Time', fontsize=16)
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.legend(loc='center left', bbox_to_anchor=(1.0, 0.5), title='Genre')
plt.grid(True)
plt.tight_layout()
plt.show()
```



- The code analyzes genre popularity trends on Netflix from the year 2000 onwards.
- The 'listed_in' genre strings are split into lists, then exploded to separate each genre into its own row.
- The data is grouped by release_year and genre, and the number of titles in each combination is counted.
- Missing values in the pivot table are filled with 0.
- The visualization reveals which genres have grown or declinedpopularity over the years. change the lanaguae of this code

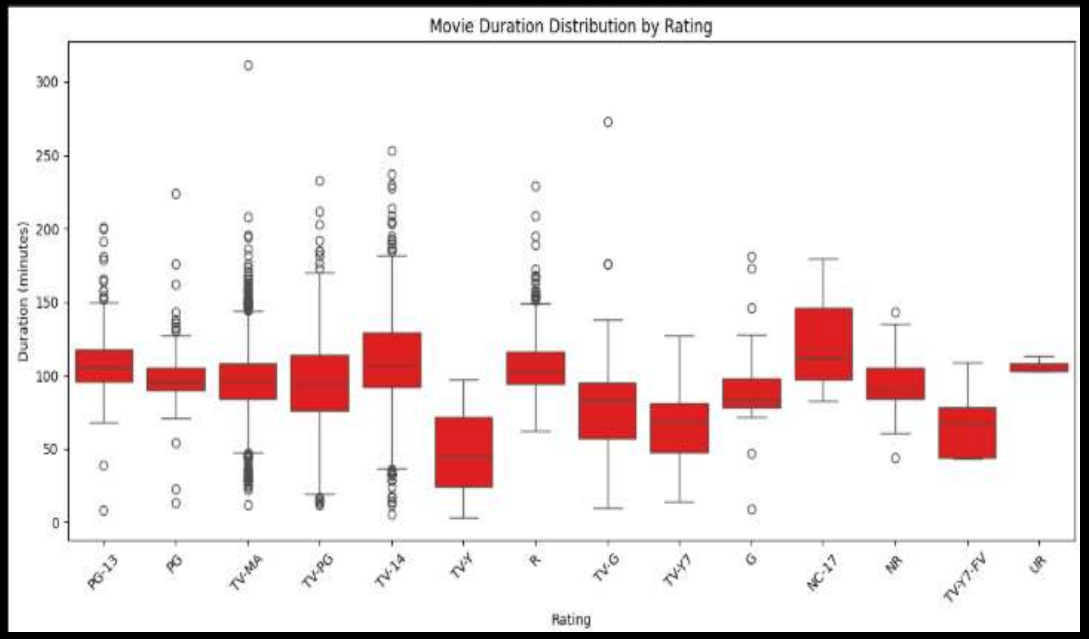
Geographical Analysis: Further explore the distribution of content across different countries and regions.

```
df['country'] = df['country'].fillna('Unknown')
countries = df['country'].str.split(', ', expand=True).stack().reset_index(level=1, drop=True).to_frame('country')

country_counts = countries['country'].value_counts()
country_counts_da = country_counts.reset_index()
country_counts_da.columns = ['country', 'number_of_titles']


fig = px.choropleth(country_counts_da,locations="country",locationmode='country names'
,color="number_of_titles",hover_name="country",color_continuous_scale="gray",title='Distribution of Content Across Countries')
fig.show()
```

- A colorful world map that shows how many Netflix shows and movies are available in each country.
- The **'country'** column in the data sometimes has more than one country listed for a show (like "United States, Canada").
- So, the code splits those entries into separate rows—one row for each country.
- It then counts how many shows/movies are linked to each country.
- Using **Plotly Express**, the code draws a map:
  - Each country is colored based on how many Netflix titles it has.
  - The **plasma** color style gives it a bright and smooth look.
  - When you hover over a country, you can see the exact number of titles.
- This map helps us understand where Netflix content comes from and which countries have the most.

Correlation Analysis: Investigate potential correlations between variables (e.g., ratings and duration).

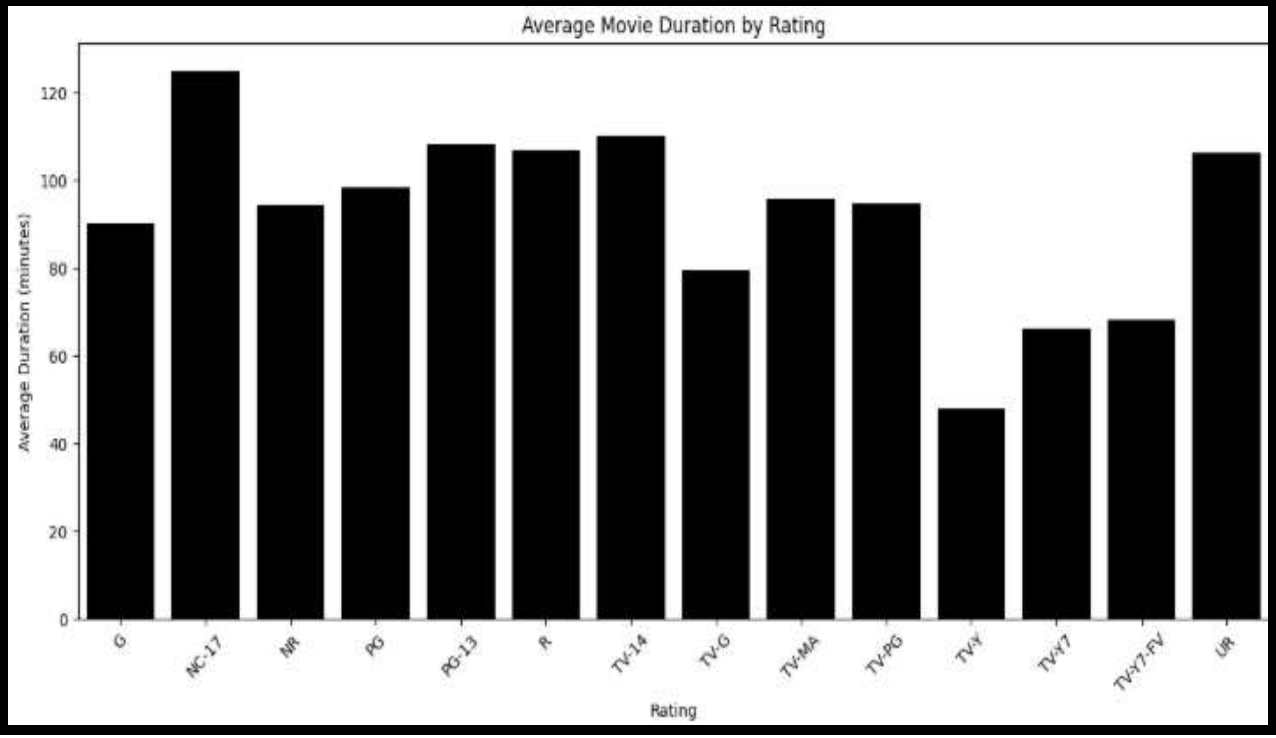
```
movies = df[df['type'] == 'Movie'].copy()
tv_shows_df = df[df['type'] == 'TV Show'].copy()

movies.dropna(subset=['duration'], inplace=True)
movies['duration_minutes'] = movies['duration'].str.replace(' min', '').astype(int)


plt.figure(figsize=(12, 6))
sns.boxplot(data=movies, x='rating', y='duration_minutes',color = "red")
plt.title('Movie Duration Distribution by Rating')
plt.xlabel('Rating')
plt.ylabel('Duration (minutes)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



- **Duration Spread Varies by Rating:**
- Some ratings (like **TV-MA** or **R**) might show longer or more variable durations.
- Others (like **PG** or **TV-Y**) tend to have shorter, more consistent durations.
- **Outliers Are Highlighted:**
- Movies with extremely short or long durations stand out as points outside the box.
- **Audience Targeting:**
- Longer movies may be more common in mature-rated categories.
- Shorter ones might target children or general audiences.
- **Helps Identify Patterns:**
- The plot gives a quick sense of how movie length relates to the intended audience (via the rating).

Correlation Analysis: Investigate potential correlations between variables (e.g., ratings and duration).

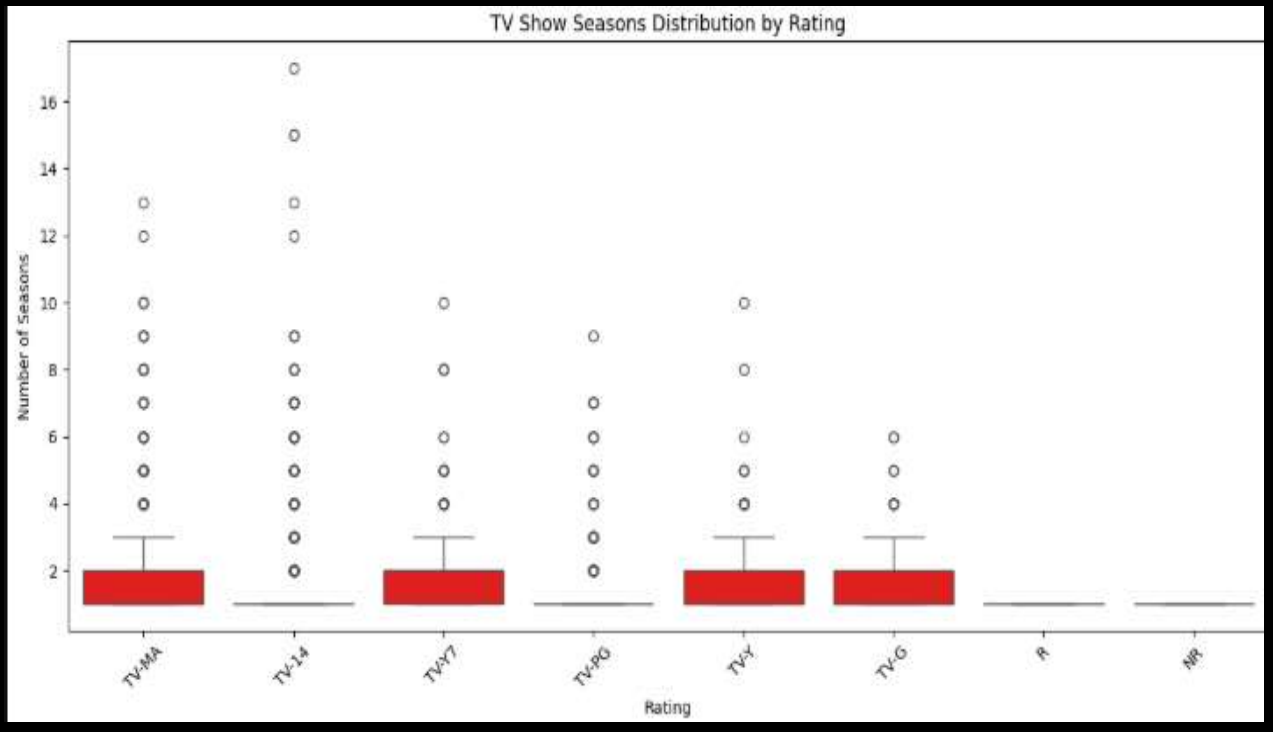
```
average_duration_by_rating = movies.groupby('rating')['duration_minutes'].mean().reset_index()

plt.figure(figsize=(12, 6))
sns.barplot(data=average_duration_by_rating, x='rating', y='duration_minutes',color = "black")
plt.title('Average Movie Duration by Rating')
plt.xlabel('Rating')
plt.ylabel('Average Duration (minutes)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



1. **Different Ratings Have Different Average Durations:**
   I. Ratings like **TV-MA** or **R** may have longer movies on average.
   II. Ratings like **TV-Y**, **TV-G**, or **PG** typically have shorter average durations.
2. **Audience Age Affects Duration:**
   I. **Adult-rated** movies are usually longer, possibly due to more complex plots.
   II. **Kids' or family movies** are often shorter to suit attention spans.
3. **Helps Content Planning:**
   I. Useful for producers, editors, or platform curators to plan content lengths based on audience type.
4. **Outliers Are Smoothed Out:**
   I. Unlike a boxplot, this shows clean **averages**, not the full spread (so it hides extreme short/long movies).

Correlation Analysis: Investigate potential correlations between variables (e.g., ratings and duration).

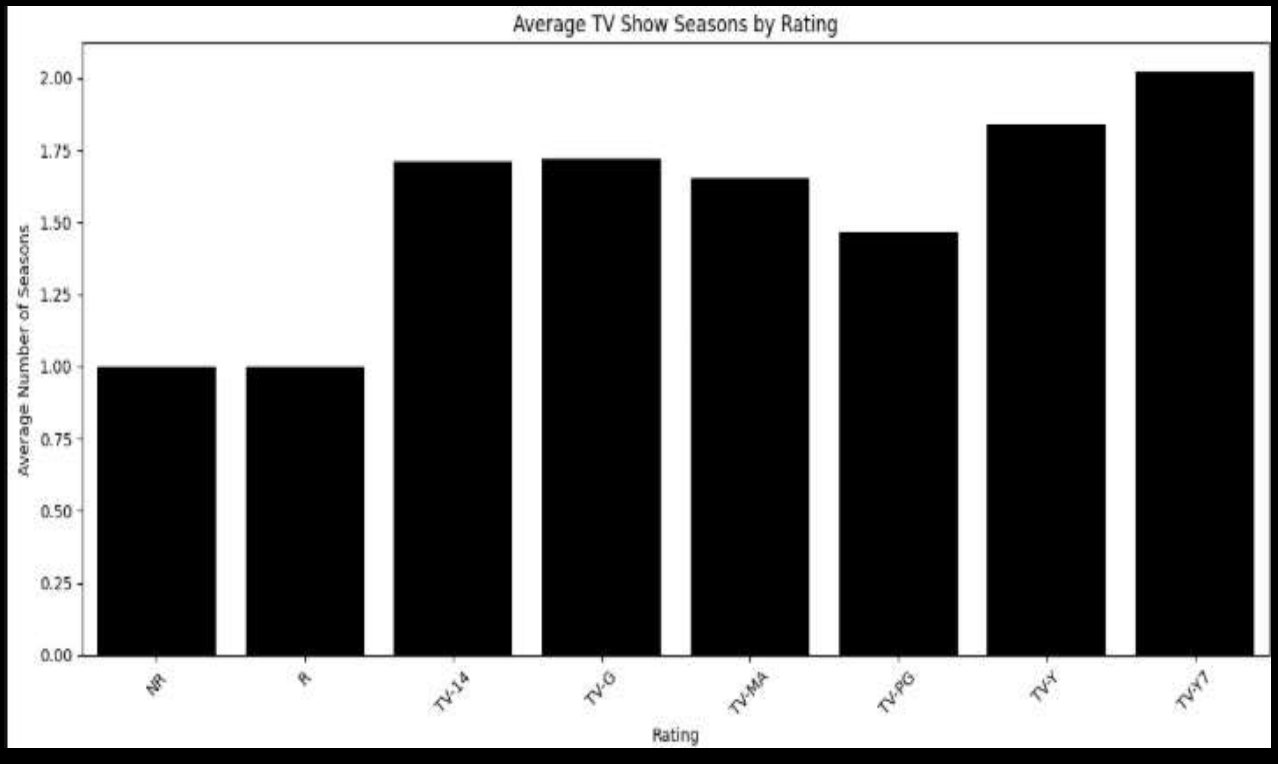
```
tv_shows_df.dropna(subset=['duration'], inplace=True)
tv_shows_df['duration_seasons'] = tv_shows_df['duration'].str.replace(' Seasons', '').str.replace(' Season', '').astype(int)

plt.figure(figsize=(12, 6))
sns.boxplot(data=tv_shows_df, x='rating', y='duration_seasons',color = "red")
plt.title('TV Show Seasons Distribution by Rating')
plt.xlabel('Rating')
plt.ylabel('Number of Seasons')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



1. **TV Show Length Varies by Rating:**
   - I. Ratings like **TV-MA** or **TV-14** may have TV shows with more seasons on average.
   - **II. Kids' ratings** (TV-Y, TV-G) usually have shows with fewer seasons.
2. **Outliers Are Visible:**
   - I. The boxplot highlights **TV shows with unusually high numbers of seasons** as dots above the whiskers.
3. **Audience Influence on Series Length:**
   - I. Shows meant for mature audiences often span multiple seasons (longer story arcs).
   - II. Shows for younger audiences are typically shorter.
4. **Content Planning Perspective:**
   - I. Helps identify how content length (in seasons) aligns with content rating, useful for content strategy or catalog analysis.

Correlation Analysis: Investigate potential correlations between variables (e.g., ratings and duration).

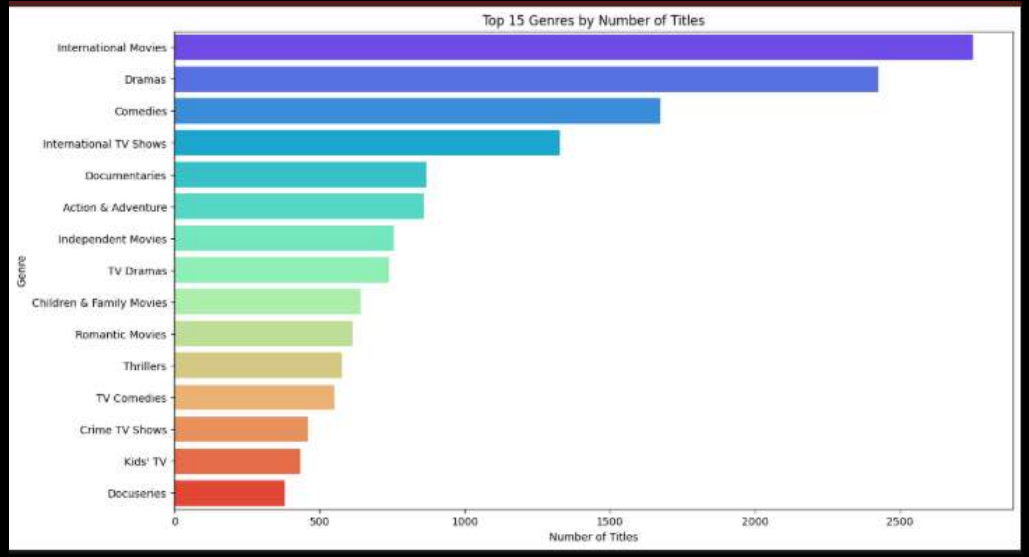
```
average_seasons_by_rating = tv_shows_df.groupby('rating')['duration_seasons'].mean().reset_index()

plt.figure(figsize=(12, 6))
sns.barplot(data=average_seasons_by_rating, x='rating', y='duration_seasons',color = "black")
plt.title('Average TV Show Seasons by Rating')
plt.xlabel('Rating')
plt.ylabel('Average Number of Seasons')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



1. **Longer Shows for Mature Ratings:**
2. Ratings like **TV-MA**, **TV-14**, or **R** often show **higher average seasons**, indicating that mature content tends to run for more seasons.
3. **Shorter Shows for Younger Audiences:**
4. Ratings like **TV-Y**, **TV-G**, or **PG** generally have **fewer average seasons**, often due to shorter attention spans and simpler formats for younger viewers.
5. **Audience Category Impacts Series Longevity:**
6. The chart suggests a **correlation between audience age group and show length** in terms of seasons.
7. **Simplified Trend Summary:**
8. This plot gives a clean view of trends without showing outliers or spread (unlike a boxplot), which is useful for quick comparisons.

Content Variety: Evaluate the diversity of content by analyzing the number of unique genres and categories.

```
genres['listed_in'] = genres['listed_in'].str.split(', ')

genres = genres.explode('listed_in')
unique_genres = genres['listed_in'].nunique()
print(f"Total unique genres/categories: {unique_genres}")
sorted(genres['listed_in'].unique())
genre_counts = genres['listed_in'].value_counts().reset_index()
genre_counts.columns = ['genre', 'count']

print(genre_counts.head(10))
df['num_genres'] = df['listed_in'].dropna().apply(lambda x: len(x.split(', ')))
average_genres = df['num_genres'].mean()
print(f"Average number of genres per title: {average_genres:.2f}")
plt.figure(figsize=(14, 8))
sns.barplot(data=genre_counts.head(15), x='count', y='genre', palette='rainbow')
plt.title('Top 15 Genres by Number of Titles')
plt.xlabel('Number of Titles')
plt.ylabel('Genre')
plt.show()
```

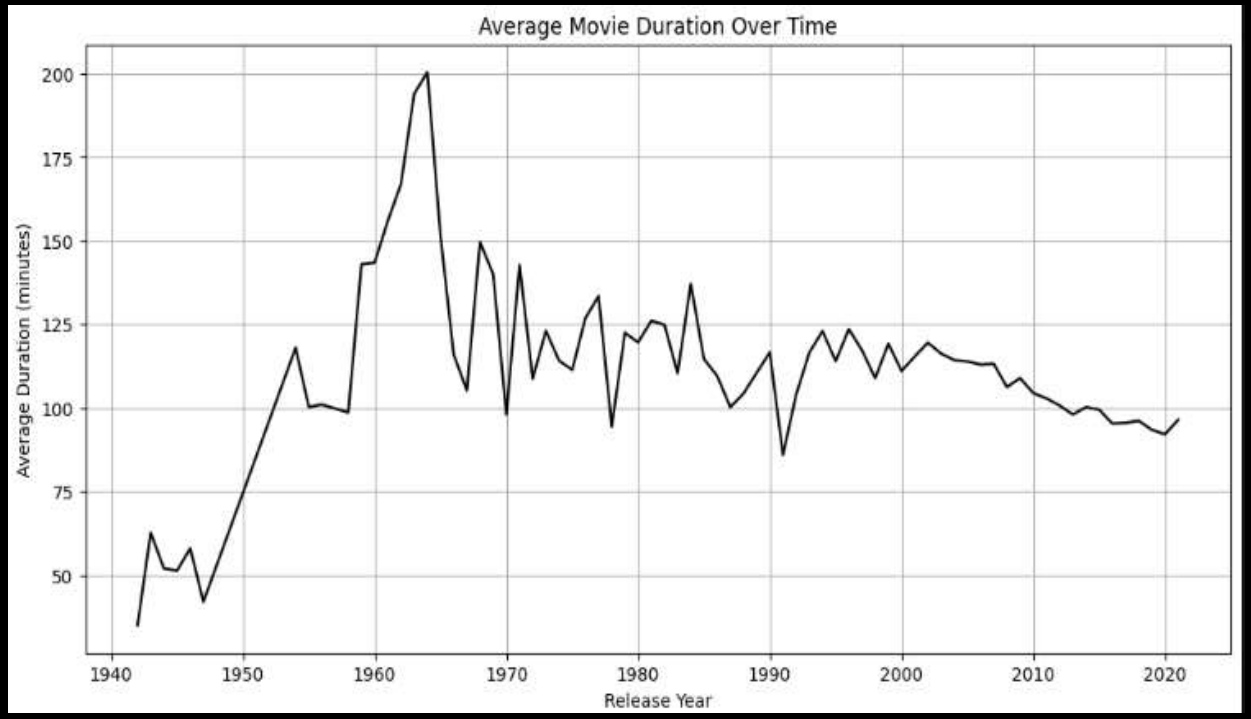



```
Total unique genres/categories: 42
                      genre  count
0      International Movies   2752
1                    Dramas   2427
2                  Comedies   1674
3    International TV Shows   1328
4             Documentaries    869
5         Action & Adventure    859
6         Independent Movies    756
7                 TV Dramas    739
8  Children & Family Movies    641
9           Romantic Movies    616
Average number of genres per title: 2.20
```

- analyzes genres/categories in a dataset (likely movies or shows like a Netflix dataset).
- It splits the listed_in column in the genres DataFrame into lists using commas as separators.
- Uses the explode() function to transform each genre into its own row for accurate counting.
- Calculates the **average number of genres per title**: Adds a new column num_genres to DataFrame a.
- Uses a lambda function to count the number of genres per title.
- Computes and prints the mean of these counts.

Content Evolution Over Time: Explore how the characteristics of content (e.g., duration, ratings) have evolved over the years.

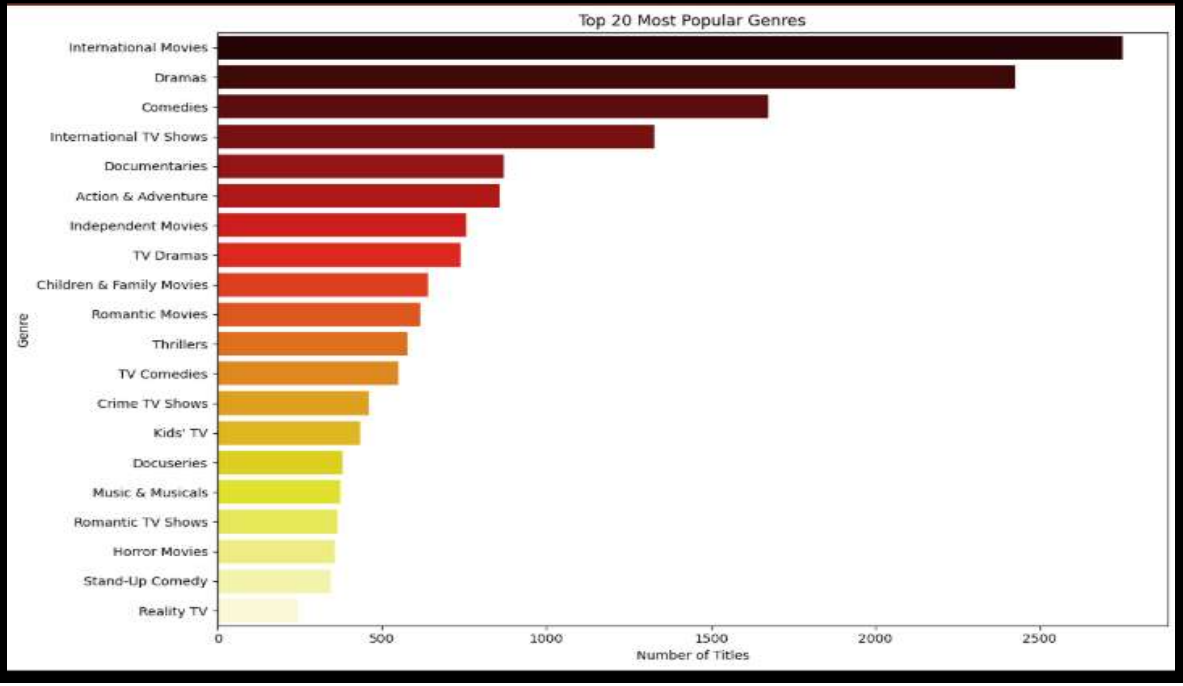
```
movies = df[df['type'] == 'Movie'].copy()
movies.dropna(subset=['duration', 'release_year'], inplace=True)
movies['duration_minutes'] = movies['duration'].str.replace(' min', '').astype(int)
average_duration_by_year = movies.groupby('release_year')['duration_minutes'].mean().reset_index()
plt.figure(figsize=(12, 6))
sns.lineplot(data=average_duration_by_year, x='release_year', y='duration_minutes',color = "black")
plt.title('Average Movie Duration Over Time')
plt.xlabel('Release Year')
plt.ylabel('Average Duration (minutes)')
plt.grid(True)
plt.show()
```



- **Trends in Movie Length Over Time:**
- The line plot reveals whether movies have been getting **longer**, **shorter**, or **remaining stable** across years.
- **Possible Industry Shifts:**
- If there's a noticeable increase or decrease in duration over certain periods, it might reflect:
  - Changing audience preferences
  - Streaming platform influence (shorter content)
  - Shifts in genre popularity (e.g., more documentaries or action films)
- **Anomalies or Sudden Changes:**
- Spikes or drops in the line could indicate specific years where content types or platform strategies changed significantly (e.g., 2020 pandemic-related production changes).
- **Modern Content Patterns:**
- Recent years might show a **decline or plateau** in average durations due to the rise of shorter, binge-friendly content formats.

User Preferences: Investigate whether certain genres or types of content are more popular among users.

```
df['listed_in'] = df['listed_in'].fillna('Unknown')
genre_series = df['listed_in'].str.split(', ')

genres_exploded = genre_series.explode()

genre_counts = genres_exploded.value_counts()

plt.figure(figsize=(12, 8))
sns.barplot(x=genre_counts.values[:20], y=genre_counts.index[:20], palette='hot')
plt.title('Top 20 Most Popular Genres')
plt.xlabel('Number of Titles')
plt.ylabel('Genre')
plt.tight_layout()
plt.show()
```



1.Most Common Genres:

1. The plot shows which genres Netflix offers most frequently.
2. Genres like **Dramas**, **Comedies**, or **Documentaries** often appear at the top.

2.Content Strategy Reflection:

1. Netflix may focus more on top genres based on viewer demand or global appeal.
2. For instance, a high number of **International Movies** or **Children & Family Movies** could signal efforts to cater to specific audiences.

3.Genre Overlap:

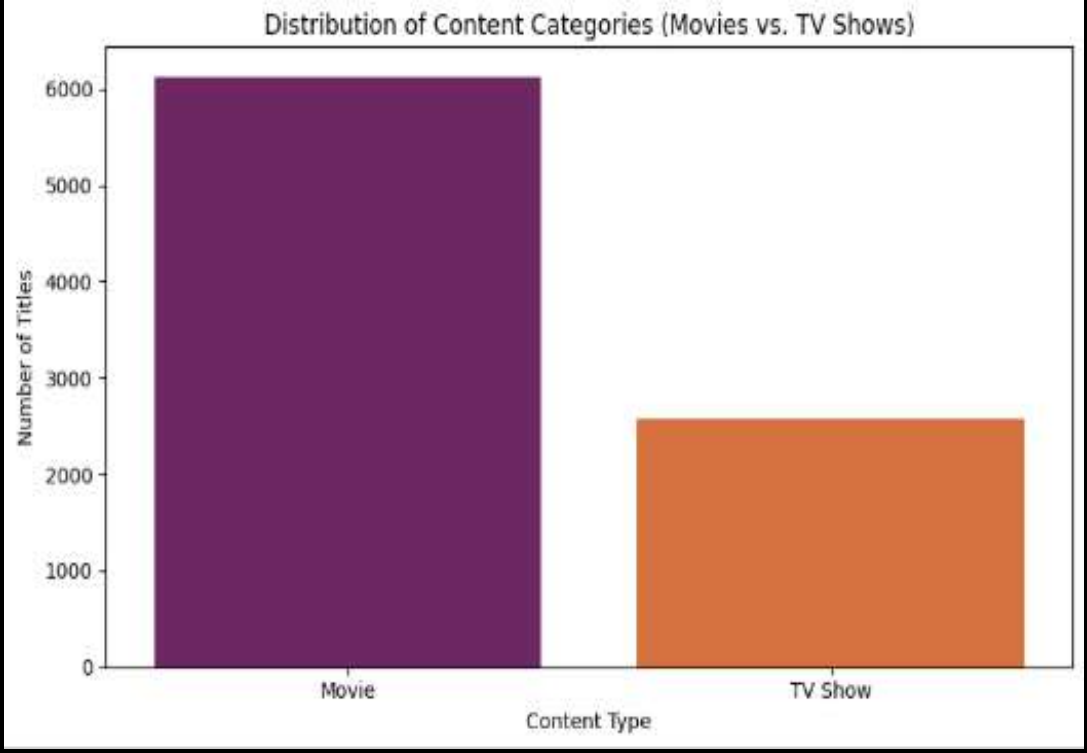
1. Many titles fall into **multiple genres**, so the counts reflect **genre appearances**, not unique titles.

4.Useful for Content Creators & Viewers:

1. Content producers can use this to align with popular genres.
2. Viewers can understand what Netflix tends to offer more.

User Preferences: Investigate whether certain genres or types of content are more popular among users.

```
category_counts = df['type'].value_counts()

plt.figure(figsize=(8, 5))
sns.barplot(x=category_counts.index, y=category_counts.values, palette='inferno')
plt.title('Distribution of Content Categories (Movies vs. TV Shows)')
plt.xlabel('Content Type')
plt.ylabel('Number of Titles')
plt.tight_layout()
plt.show()
```



- **Movies Likely Dominate:**
- Typically, the plot shows a **higher bar for Movies**, meaning Netflix has more movies than TV shows in its catalog.
- **Content Strategy Snapshot:**
- A higher number of movies suggests Netflix may focus more on film content quantity.
- If TV shows are close in number, it reflects a balanced strategy between long-form series and standalone movies.
- **Quick Overview of Catalog Mix:**
- Helpful for understanding how diverse the platform is in terms of format.
- Creators and marketers can gauge what format is more saturated or competitive.
- **Foundation for Deeper Analysis:**
- Can lead to more detailed breakdowns (e.g., by year, genre, region) to understand **how the balance shifts over time.**

Conclusion

The Netflix dataset analysis gave a comprehensive look into the streaming platform's content distribution. The data showed that movies dominate over TV shows, the U.S. leads in content contribution, and genres like dramas and international content are most common.
The cleaning and transformation of the dataset prepared it for meaningful visual storytelling.
This project was a valuable exercise in real-world EDA, data cleaning, and using visualization to tell compelling data stories.

- Netflix content surged after 2010.
- TV Shows and Movies have different duration and rating patterns.
- Genres like Drama, International, and Documentaries are most popular.
- USA and India lead in content contribution.
- Content ratings and duration show evolution over time.

References

Tools: Python (pandas, seaborn, matplotlib), Jupyter Notebook

Dataset:"C:\Users\dell\Desktop\neetant\netflix_titles.csv"

Linkidin:www.linkedin.com/in/neetant-meena-8a60a1293

Csv.file:"C:\Users\dell\Desktop\neetant\.ipynb_checkpoints\netflix-checkpoint.ipynb"